ESD-TDR-64-168 SR-124

SYSTEM DESIGN AND ENGINEERING FOR REAL-TIME MILITARY DATA PROCESSING SYSTEMS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-168

JANUARY 1965

D. R. Israel

Prepared for

416 L/M/N SYSTEM PROGRAM OFFICE
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 416L

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
AF Contract 19(628)-2390

ESD-TDR-64-168 SR-124

SYSTEM DESIGN AND ENGINEERING FOR REAL-TIME MILITARY DATA PROCESSING SYSTEMS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-168

JANUARY 1965

D. R. Israel

Prepared for

416 L/M/N SYSTEM PROGRAM OFFICE
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 416L

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
AF Contract 19(628)-2390

ESD-TDR-64-168 SR-124

SYSTEM DESIGN AND ENGINEERING FOR REAL-TIME MILITARY DATA PROCESSING SYSTEMS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-168

JANUARY 1965

D. R. Israel

Prepared for

416 L/M/N SYSTEM PROGRAM OFFICE
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 416L

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
AF Contract 19(628)-2390

FOREWORD

This is to acknowledge the significant comments and assistance from a host of colleagues at MITRE during the preparation of this paper. It is our collective experience which forms the basis of this presentation.

NOTE: The material in this report was originally prepared for presentation at the SHAPE Symposium on Computer Programming for Military Systems (September 21-25, 1964) at The Hague, Netherlands.

ABSTRACT

This report treats the key problems and considerations arising in the design, engineering, and implementation of military systems in which real-time data processing plays a central role. The principal distinguishing characteristics of these command and control systems are summarized. Organizational matters relating to responsibilities, operational inputs, and procurement aspects are described in the context of the over-all system acquisition process. Initial considerations which should guide the over-all design are discussed, including such outstanding design problems as the proper matching of man/machine capabilities and the provision of adequate capacity and flexibility for change and growth. Important aspects of hardware, software, and testware design are also detailed.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved.

John E. Plevin Maj
for JOHN A. TRASK
Col., USAF
System Program Director
416L/M/N System Program Office

TABLE OF CONTENTS

INTRODUCTION

This paper addresses itself to some of the key problems and considerations arising in the design, engineering, and implementation of military systems in which real-time data processing plays a central role. These are the so-called command and control systems. To be more specific, I have in mind something like the NADGE system; in U.S. experience, examples would be SAGE, 412L, or the NORAD COC.

I do not believe it is possible -- or even useful -- to establish a completely satisfactory definition of this class of systems. Instead, and as a means of establishing a basis for the subsequent remarks, let me summarize some of the principal characteristics of these military data processing systems:

(1) They are generally large, as measured in terms of equipment or geography, and usually very costly.

(2) They are typically made up of many elements or subsystems -- computers, displays, communications, personnel, computer programs, data bases, input data sources, and output data users -- all of which must be carefully integrated.

(3) They process large amounts of raw data from diverse sources, converting this to summarized information for men or other machines or systems.

(4) They must perform the data processing in "real-time" in order that the system response keeps pace with incoming data and required outputs, and they must be available on a continuous, around-the-clock, basis.

(5) While they normally operate well below their design capacity, a capacity which may never be required except in wartime, they must operate continuously at a high level of readiness.

(6) They usually replace an existing manual system and, in turn, must operate as a part of some larger military system.

(7) Despite the introduction of automatic data processing, human operators have a predominant role in the system operation.

(8) They are generally designed and implemented by one organization for use by another.

(9) Because of their size and importance, their development is usually marked by a large number of approval levels and coordination channels.

(10) They take a significant time to acquire, during which time both the requirements and available technology may change.

(11) They must change and evolve during their operational lifetime to meet new requirements and situations.

(12) They usually defy an *a priori*, detailed, quantitative, specification of their performance because of their complexity and the unpredictability of the conditions under which they operate.

We are considering, then, a class of systems which differs significantly from the more conventional application of computers to scientific problems or business data processing. For the most part, these command and control systems are of a complexity beyond our normal engineering experience. In view of these enumerated characteristics, it is not surprising that successful system development and implementation -- as measured in terms of expected cost, schedules, and performance -- has, to date at least, been more the exception than the rule.

I should also like to avoid defining the terms system design and system engineering. I will use system engineering as the broader activity including such functions as design, technical support of procurement and production, and testing. The significant point to be made is that system engineering must include a host of non-technical problems if successful system development and operation is the goal. The system design and engineering plan must be an appropriate balance or compromise among the factors of operational requirements, technological capabilities, costs, and schedules.

The system engineer should take an extremely broad point of view. His scope must cover the long time span stretching from initial concept to ultimate operation, and even beyond. These systems cannot remain static, but must evolve to meet new requirements. His considerations should cover a diversity of topics. It is unwise, if not dangerous, to consider only the equipments and computer programs directly related to the operational functions. Proper system engineering will consider such varied topics as availability, training, and exercising of operators; operational procedures; provisions for the maintenance of hardware and computer programs; the availability and operating condition of government-furnished equipments; communications integration with other systems; civil engineering and construction of buildings and related facilities; and so on.

Beyond this, it is increasingly evident that successful system development is heavily dependent upon the management techniques and procedures used to guide the system from concept to field operation. Or stated somewhat differently, it is heavily dependent upon the design of the acquisition process or acquisition system.

It is in this broad context of topics stretching over a long time frame that I will discuss some of the design and engineering problems of command and control systems. I will draw heavily from some ten years of experience in the United States, during which time we have almost completed the development of a first generation of such systems. I will cite some instances from this experience where possible.

As a final introductory remark, let me say that there are no known magic formulae, no known optimum procedures or techniques, nor any ten easy lessons to successful system design and engineering. My principal intentions are only to point out items which should be considered and to illuminate possible pitfalls. Much of what I will say may strike you as little more than common sense. If so, I can only wish that our foresight had been as good as our present hindsight. We have learned much about system engineering over the past decade; we have learned much less about how to document or apply this experience. I am quite conscious of the difficulty of transferring such experience, but I hope the following remarks can in some way be of benefit to the engineers and designers of future systems.

ORGANIZATION

In keeping with the announced broad perspective, let me first discuss some general organizational matters relating to responsibilities, operational inputs, and procurement.

Project Organization

It is generally agreed today that a strong, centralized, knowledgeable project organization should be established by and within the government agency with the primary responsibility for system development. This seems to be necessary regardless of the type of contractual responsibilities and arrangements made -- prime system contractors, associate contractors, or equipment suppliers.

This project organization should have a broad charter, both in scope and time. It must consider the relationship of such problems as construction and facilities, training, maintenance, and exercising to the over-all design since they are likely to exert a strong influence on the system and its eventual performance. It should be charged with coordination of all aspects of the design. It should review and approve all schedules, monitor progress, provide a central and common focus for all system decisions, and exercise the necessary resource management.

This project organization should have a single, enthusiastic, strong-minded leader. It should not be headed by a committee which does not have requisite authority or which can only arrive at decisions by finding the least common multiple.

The project organization should be directly supported by technical competence equal to the job at hand. This may vary with the contractual arrangements, but in no case should the project organization find itself incapable of dealing on an adequate technical basis with the contractors.

I must stress the project nature of this organization. It should be organized for a specific purpose and given the necessary authority and responsibility. It is the antithesis of the stable engineering organization formed along conventional functional lines. An attempt to split up the system engineering job to satisfy some existing functional organization should be avoided since it is likely to jeopardize the entire endeavor from the outset.

Operational Inputs

The arrangements for the project organization should assure early, continuous, and active representation from the ultimate using and operating command.* This is a requirement in the design process for the following reasons: first, it will provide a direct channel of information concerning changing requirements; second, it will provide the necessary contact with field operating problems; and third, it will provide mutual appreciation of problems by the developer and user.

There is a strong and very dangerous tendency for an engineer to design a system for himself to operate, and generally to operate only for short periods of time and at high, or at least interesting, load levels. The actual operators, on the other hand, are less sophisticated technically and usually must operate the system day-in and day-out for eight-hour periods at very low and uninteresting load levels -- levels at which the automatic system generally is not required.

Put in somewhat cruder terms, the engineer tends to design a system which is difficult to operate or which only he can operate. (On the other hand, the operator tends either to redesign the existing system

---

*In some situations, the using and operating commands may not be the same. This distinction will be ignored here and the terms <u>user</u> and <u>operator</u> will be used interchangeably.

or to design a system which cannot be built.)

The solution is close participation by the using command throughout the design phase. The participation must be from properly chosen personnel and their inputs must be controlled so that the design does not become merely a "wish book" which cannot be met within the constraints of the allocated resources. The participation should be continuous if waste is to be avoided. Confrontation between the operator and developer at six-month or longer intervals leads to misunderstandings, unnecessary effort, rework, and often unsatisfactory compromises resulting from tradeoffs occurring too late in time. The operator should be required to concur on all operational design features, particularly those relating to operational personnel and man-machine relationships.

The representatives from the using command should include both planners and operators, with the balance gradually shifting to the latter as the project matures.

The assignment to this function of development personnel with operational experience or personnel who were previously assigned to the operating command is a poor substitute for direct representation from that command. Such representation has two bonus effects. First, it provides for a direct association of the operating command with the system development, and thus gives it an opportunity to be party to design and development decisions as they are being made. And secondly, it may help to alert and prepare the operating command for actually using the new system. This command is generally over-committed to the operation of an existing system and finds it difficult to prepare -- for example, with operating procedures and manuals -- for operation of the new system.

Procurement Aspects

While it is not the purpose of this paper to explore in detail the special procurement and contractual problems associated with military real-time data processing systems, several related problems and considerations are of direct interest to the system engineer:

(a) The desirability of a prime contractor as opposed to associate contractors is still being debated, and I will not try to elaborate on the advantages and disadvantages of each form. It should be pointed out, however, that in both cases, and especially with the latter arrangement, the requirement still exists for a strong in-house technical and managerial capability.

(b) In either contracting situation, experience points out the desirability of a separate, independent test contractor, who is not responsible for the original design or any of its implementation, to help assure that the system is properly delivered, installed, checked out, and evaluated.

(c) Computer programs -- the "software" -- can be acquired in several different ways. Software manufacturers not involved with equipment production are becoming increasingly available. Software can also be secured from equipment manufacturers, but the use of a company different than that which supplies the system computer can introduce some difficulties relating to release of proprietary information. Generally, all of the hardware should be procured first, since the specific hardware choices will affect the magnitude and complexity of the software.

(d) Software is difficult to procure under fixed-price contracts due to the difficulty of producing detailed performance specifications for these programs. As opposed to hardware, software specifications are always in the language of the user rather than the supplier. As with hardware, cost should not be the primary or only selection criterion for the software contractor. Experience, past performance, geographic location, and availability of trained manpower should weigh heavily in the ultimate selection.

(e) Software procurement should be made on the basis of a plan which considers the longer-term software problems. Will the military itself take over software production at some point in time?

(f) Caution should be exercised with hardware procurements based on detailed design specifications unless the procuring agency has assured itself that a product which meets these specifications will provide the desired performance. This is very difficult to do.

(g) Caution should also be exercised relative to advertised "off-the-shelf" availability. Due to the long lead time of computer programs, an early delivery of the computer is essential. Accordingly, a 120-day availability of a computer with the same characteristics as the ultimate production models might be an adequate proof of being "off-the-shelf."

BROAD OR CONCEPTUAL DESIGN

This section discusses some of the initial considerations which should guide the over-all design. Detailed aspects of hardware, software, and testware design follow in succeeding sections.

What Is The System?

Automated command and control has become extremely fashionable in recent years, achieving some appeal as a military status symbol. This, among other reasons, can lead to the initiation of a system development without a full understanding of what the system is and what it is expected to do. When this happens, the designer, the user, and the taxpayer may later regret the initial precipitate action.

Too often we hear the designer lament the fact that the user cannot supply him with an adequate statement of requirements. When this is said, it usually means that the designer doesn't understand the nature of these systems or the military problem. In practice, the user's requirements are not easily expressed in quantitative terms, and they can only be put into meaningful form when matched with the available technology and possible designs. In many cases, the available technology identifies or even generates the requirements.

Thus, as the very first step in the design effort, the designer and user must come to an understanding of the military problem, and they should prepare and document a matched statement of requirements and the corresponding conceptual or broad design. In doing so, the full implications of the system and its operational use should be explored. Both parties should consider the automated portion of the system as an iceberg, and a significant initial effort should be directed towards understanding and describing what is under the surface.

For example, the implications for manning, training, maintenance, and logistics must be fully understood by the user. Too often we are disappointed by initial claims of savings of operational personnel. I know of no system in which actual personnel savings have resulted, although in most cases the capability (or productivity) of the operations personnel has been greatly increased.

A related question requiring an early answer is: "What constitutes the system?" In particular, what equipments already in use by the manual system are to be employed? In determining this, the operating condition of these field equipments should be realistically ascertained. The early development of SAGE was hampered by the fact that the radars were not considered as a part of the system. In both SAGE and 412L, the actual operating condition of such field equipments was not properly understood, and difficulties resulted.

A careful inventory and field survey is recommended. This applies to both equipments and facilities. The system designer, often an electronics engineer, is quite prone to ignore the seemingly prosaic problems of roads, buildings, power, and air conditioning. Measurements on field equipments under field maintenance are advisable. A radar under field conditions performs quite differently than at the manufacturer's plant. SAGE and 412L experience attest to this.

At this early phase, the interfaces with other systems -- present and future -- must be considered. An exchange of data with other systems is almost always required, and the problems of data compatibility-- in terms of information content, format, and signal levels -- must be solved. Not only should these problems be considered at the design level, but responsibility for any changes to achieve compatibility must be clearly assigned. The NATO efforts in establishing data link standards and formats are to be commended; it is a step which has been badly ignored in the U.S. to a point where it may be too late to create meaningful standards.

Briefly then, at the outset there should be a careful consideration and documentation of what the system is, what it will do, what comprises it, and how it relates to other systems.

Evolution

As noted earlier, significant periods of time are involved in the acquisition of these systems. When they do become available, they may not meet the then current requirements and may be very difficult -- in terms of time or effort -- to modify so as to achieve the desired goals. This has given rise in recent years to a general feeling that the design and implementation of these systems should be pursued on a more evolutionary basis.

The merits of an evolutionary approach are said to be that we can proceed in small, sure steps. It would provide for an early demonstration of capability and would ensure operational experience and feedback before proceeding with a more sophisticated design. Headquarters command systems are very strongly influenced by the personality and desires of the commander himself, and an evolutionary capability would make it possible for individual commanders to tailor or modify the system accordingly. It is further felt that the sophistication and technical difficulties which have characterized the first generation of these systems could be avoided in the future if there were greater use of "off-the-shelf" equipments and a more direct initial mechanization of existing manual operations.

The current emphasis on evolution, however, may confuse what are perhaps two separate ideas. The first is the concept of a time-phased implementation, or what might be termed a planned evolution towards a predetermined design. The second relates to provisions for the unplanned modifications necessitated as a result of operating experience or changes in requirements or the technology.

In this light, the evolutionary approach has some dangers which should not go unrecognized. In particular, a time-phased implementation should not be used as an excuse for avoiding the total system design problem. If only the first increment is planned, then there is a strong possibility that inadequate attention will be given to the design requirements imposed by subsequent steps. Specifically, the universal lack of funds and the influence of ever-present economy drives cause strong pressure in this direction and may force a decision to buy only the equipment or capability required for the first steps and not allow time for sufficient analysis and design of the total, long-range system. Restrictions on building size, power, or air-conditioning can be as serious as limited computer capability. Subsequent steps are then very difficult and costly to implement, possibly requiring grossly different equipments and costly retrofits to existing equipments.

Another danger of time-phased implementation is that if the subsequent steps are not very carefully planned, major problems can arise as these steps -- involving additional programs and possibly equipments -- introduce interference with the operation of the system. A related difficulty is the retraining of operators. In fact, there appears to be a critical mechanization level which should be incorporated in an initial step. This should include the full mechanization of the essential inputs and outputs, and should include the majority of the operator facilities (consoles, keyboards, switches, etc.), even though all of this capability might not be used initially.

It should further be noted that requirements for testing are such that initial system testing is not in any large percentage salvageable for subsequent steps, and a complete series of tests with all attendant costs may be required for the replacement or next model of the system.

Another point is that while direct mechanization of existing procedures may yield an early demonstration of progress, it should be applied with caution as it may prevent the realization of a vastly improved operation which could be achieved by fully exploiting the automatic data processing capability. Neither the automobile nor the airplane were products of direct mechanization.

There is also the question of the use of "off-the-shelf" equipments. This, too, must be approached with caution and reason. There is very little that is actually "on-the-shelf", particularly in the sense that it can be applied directly to a system. Further, it is extremely important to differentiate between "off-the-shelf" and "in the brochure".

Design guidance in this broad area, then, would seem to be as follows:

(1) The design should contain a reasonable degree of excess capacity and flexibility in all subsystems and elements to permit long-term unplanned evolution. This excess capacity must then be carefully controlled so that it does not dwindle away before it is needed.

(2) If the system is to be implemented in steps, the design must consider the resulting system first, allowing sufficient capacity to achieve this goal. Only when this end design is well understood should the steps in the implementation be delineated.

(3) The first step should not be too big to prevent an early

demonstration of progress, but should be sufficiently large so that it includes those features strongly affecting operator actions and activities.

(4) The design of a step-wise implementation must not ignore the problems of disruption and degradation of current operations.

(5) The designer should consider with caution the early benefits of a direct mechanization and the claims for the applicability and availability of "off-the-shelf" equipments.

Degree Of Automaticity

The proper degree of automation to be provided in most command and control systems is not easily determined. Many system functions are routine, easily-described, repetitive data processing tasks which lend themselves directly to mechanization; other functions clearly require sophisticated choices or decisions which can or should involve human judgment. Unfortunately, however, a large number of important functions usually cannot be so easily categorized. Here the cost and complexities of automation must be balanced against the problems and difficulties attendant to human operators.

First, the man-machine relationship is severely limited by the cost and complexity of the devices -- consoles or large-screen displays -- available for the information exchange. The operator can only do a satisfactory job if he is given the necessary data upon which to make his decision. This is suprisingly difficult and costly to accomplish, both in terms of hardware and software.

Second, the involvement of an operator introduces both equipment and human reaction times. Delays can arise both in the computer program and in the display devices. Consoles may introduce several seconds of delay depending on the storage medium which drives them; large screen displays may involve processing and mechanical transport delays of ten seconds or greater; keyboards provide a flexible input capability, but require significant times for message composition.

Third, the operator may not have the information required to make a better decision than the computer. In SAGE, it was felt that whenever the automatic tracking program got into difficulty as a result of too little, too much, or ambiguous data, the operator would be alerted to correct the situation. An evaluation of one automatic track monitoring scheme later showed that the operators did not, on the average, improve the situation over what the computer would have done if it had been left alone.

The other side of trade-off is that the automation of complex decisions can be extremely difficult and costly in computer capacity. A satisfactory automatic weapons assignment program must consider a multitude of factors: position, heading, and altitude of bomber aircraft; position and importance of possible targets; locations, status, types, and capabilities of available weapons; and so forth.

Other decisions are not as simple to automate as they first appear. Early in the system design there is a disturbing tendency to oversimplify and underestimate, and we are subsequently embarrassed by the computer time or storage consumed by what had been felt to be a simple process.

Let me mention two examples from SAGE. The first is automatic initiation of tracks from radar replies reported by a scanning radar. This was first said to be a simple pattern recognition process: look for radar replies in close proximity on successive scans, allowing some leeway for tracks with low blip-scan ratios. In actual practice, a high level of noise replies requires a very sophisticated process if the wholesale generation of false tracks is to be avoided or if the initiation of real tracks is not to be overly delayed.

Track-while-scan of aircraft is still dismissed by many people as a simple process. Even if they include such considerations as low blip-scan ratios, false tracks, and turning tracks, they generally have ignored the problems of proximate or crossing tracks. This involves another level of design sophistication.

In both cases, it wasn't until a rather detailed knowledge of the actual physical situation became available and extensive computer programming had been performed that a full realization of the cost and complexity was reached.

Finally, decisions as to the degree of automation must be carefully made in light of operator limitations and training problems. The danger of the engineer designing the system for himself to operate is ever present. The performance of many of our automated systems -- SAGE and 412L are examples -- is overly sensitive to operator qualifications and training. This, in turn, places a very high premium on provisions for continuous exercising and evaluation of these operators. Such exercising and evaluation, as is discussed next, is not without its own costs and problems. An even better solution, where possible, is to provide the system with a meaningful day-to-day job. The integration of air defense and air traffic control has always seemed to be a natural combination. It is a pity that political and other constraints seem to prevent this union.

Self-Exercising Capability

As previously noted, during peacetime these systems normally operate at rather low load levels, far below design capacity. Since they are quite sensitive to operator performance, the best possible system performance will be attained when required only if a self-exercising capability to maintain and improve operator proficiency has been included as an integral part of the over-all design. Such a capability, by which the system can be subjected to simulated high loads and special input conditions, will also permit on-going evaluation, checks on system performance, and those demonstrations of system performance which are often required for visitors.

As a part of the system design, then, careful attention should be given to the questions of which environmental conditions and sources of input data should be simulated and how this can best be done. Additional equipments are generally required for this purpose.

If the system has many sources of data, the generation of consistent inputs relating to the same environmental situation may in itself constitute a difficult task requiring data processing equipment. For example, in an air defense system it is desirable to simulate enemy bomber flights as seen by a radar. It is possible to do this in real time by suitable analog or digital simulators at a radar site, or the simulated radar returns can be prepared and prerecorded on a magnetic tape or photographic film which can be scanned by appropriate equipment at the site. Either of these techniques would be suitable for simulation at an individual site. However, if the system consists of many radars with overlapping coverage and the aircraft will traverse the coverage of several sites, it becomes necessary to simulate a consistent set of radar inputs. This generally requires the coordinated generation of tape or film inputs. This, in turn, entails a significant data processing task in the preparation of aircraft flight paths and the computation of the r, θ data as observed by the radars. A general-purpose computer may be useful or necessary for this purpose.

In any event, it is generally advantageous for training purposes to arrange the simulation so that it is repeatable and situations can be easily reproduced for the same or different sets of operators.

A second problem of exercising is that of concurrent operation with live inputs. Even if it is electronically possible to mix the live and simulated inputs, this may cause extreme confusion and danger unless the computer has some means of segregating and identifying each. This may require excess computer capacity for the extra processing and display programs required to differentiate between the simulated and live inputs, at least to selected operating personnel.

Some simulated inputs cannot be preplanned since they are affected by the system operation and may require real-time data processing and human decisions. During a SAGE training exercise in which a center is removed from the air defense net, voice radio vectoring instructions are monitored by simulated pilots who insert the directed values of heading, speed, and altitude into the computer by special switches. A special computer program uses these inputs to simulate the flight paths of the interceptors. The current positions of these aircarft are then determined, permitting the computation of the simulated radar returns. Thus, system exercising involving non-preplanned, dynamic inputs may require special input facilities, added operating personnel, and additional data processing capability.

Finally, adequate exercising facilities must also provide the means for evaluating the system performance. Rapid feedback to operational personnel is a requirement if they are to understand and correct errors. Depending on the situation, real-time evaluation or post-test analysis of recorded data can be employed. This evaluation and analysis is not without cost, facilities, and design effort.

It should be noted that the facilities required for system exercising are closely related to those required for program and system checkout, shakedown, and test. The facilities should be coordinated and, where possible, combined.

Performance Monitoring

As systems become larger and more complex, and particularly as the number of input and output subsystems grow, it becomes increasingly difficult to determine whether the over-all system is operating properly and to isolate the causes of difficulty. Accordingly, suitable provisions for performance monitoring, trouble detection, and quality control should be included as part of the system design. The requirement for continuous system operation, coupled with the relatively abundant opportunities for subsystem malfunction, means that the designer should not expect that performance checks during preventive maintenance periods will suffice. He must consider the need for dynamic or real-time performance monitoring and diagnosis of malfunctions.

The central computer and the computer program are not major problems in this regard. The majority of their random or intermittent malfunctions can be detected by parity checking, and in many cases an immediate repeat of the computer or program operation can be successfully conducted. Other errors generally cause an obvious system malfunction or complete stoppage. However, the input and output subsystems offer many opportunities for miscalibration, random errors, poor performance, and complete outage in a fashion that does not cause the system to cease operation. Individual communication links, particularly those carrying digital data, can become completely inoperative; individual radars can go out of calibration or can fail to report targets entirely; or individual consoles or manual input devices can fail entirely -- all for significant periods of time before the degradation or failure becomes apparent to operating personnel.

Most subsystems have some "built-in" performance monitoring: parity alarms, power lights, etc. However, in many instances the subsystem equipments, their failure indicators, and those who maintain these equipments are remotely located, far from the eventual users of the data which they provide. Experience indicates that adequate attention to the subsystem performance may only occur when the human element comes into play -- specifically, when the user complains. Hence an objective of the system performance monitoring should be to provide the user with the tools and information to complain accurately.

The designer should consider the automatic and periodic generation of test messages that can be routed through these subsystems in a systematic manner for checking purposes. When difficulties are encountered, more specific and detailed check messages or techniques can be called upon to isolate the particular source of the difficulty. In many cases, it may be possible to correct or bypass the source of the difficulty temporarily until it can be fixed. Finally, a continuous summary of the status of the key elements of the system should be made available to both operating and maintenance personnel.

Two examples from SAGE might help to illustrate the general ideas. In the first, the radar data processing devices at each radar site were designed with a provision for periodically introducing false targets at predetermined range and azimuth positions, and the central computer isolates and processes these messages to check a number of the processing and communication facilities between the radar and the central computer.

The second example relates to the importance in the SAGE design, as with other netted radar systems, that the radars be accurately aligned in both range and azimuth. If not, the generation of multiple data trails and false tracks for a single aircraft may result. Since both the radar and radar data processing equipments can become misaligned during maintenance on the antenna, decoders, etc., a performance monitoring feature was added to the SAGE computer program. With the feature, the computer checks reports from different radars on the same aircraft and determines if a better match of incoming data would exist if bias errors are assumed in the azimuth data. If it is found that an azimuth correction on a radar improves the consistency of multiple radar reports, this error value is then introduced before processing subsequent reports. The error is also printed out for corrective action at the radar site at the next maintenance period.

Performance monitoring, then, should be recognized as a system function on a par with the more familiar operational functions. It should not be allowed to slip into the background during the system design. Utilization should be made of the performance monitoring built into existing equipment subsystems, but added hardware and software may be required and this should be reflected into the design of system equipments, computer programs, and communication links.

Continuity and Modes of Operation

It will be desirable to subdivide this topic into two parts. The first part deals with continuity of automatic system operation under normal or peacetime conditions. The second part relates to alternate modes of operation, generally at lower capacity and capabilities, when key elements of the system have been put out of operation. For ease of reference, normal operation will be referred to as Mode I, an alternate automatic arrangement as Mode II, and a completely manual backup as Mode III.

Failures of critical system elements will cause an interruption of Mode I operation unless adequate design measures have been taken. The usual solution is the provision of duplicate or duplex equipments coupled with error detection facilities and a rapid switchover capability. In the case of one-of-a-kind equipments, full duplicates would be required; in the case of a multiplicity of units -- memory units, display consoles, tape drives, etc. -- only a few spare elements might be needed. In the case of consoles, a general-purpose console design rather than special-purpose consoles designed specifically for an operational position should be considered. With suitable program parameter and switch label changes, it is possible to adapt a general-purpose console to individual positions, thereby retaining desired flexibility and permitting rapid replacement or substitutions in event of console failures.

With improvements in equipment reliability, provision of duplicate units, and suitable design, unscheduled interruptions of Mode I continuity can be brought down to whatever low level is desired. The designer must not overlook, however, requirements leading to scheduled interruptions of Mode I continuity. These include the functions of maintenance, equipment retrofit, program retrofit, and possibly system exercising.

Maintenance is self-explanatory. Equipment and program retrofit may result from initial design shortcomings or from the evolving nature of the system. Program retrofit must include a thorough checkout on the actual machine and is not as simple as merely changing a tape unit and reading in a new program. As noted earlier, system exercising may require interruption of normal operation unless the live and simulated inputs are properly handled.

It is important not to underestimate the amount of downtime required to perform these functions. This is particularly true in the early months or years of operation where several hours may be required each day. On a long-term basis, daily maintenance or exercising may still be required, with only occasional changes to the hardware or software.

A spare or duplicate unit for each type of element in the system may permit continuity of Mode I operation during limited types of equipment maintenance and retrofit. A full duplex would permit a Mode I operation while performing any of the functions. Reverting to Mode II or Mode III operation is also possible, although the latter is not very desirable.

Duplexing will not, of course, prevent physical destruction of the key elements of the system and the consequent interruption of Mode I operation. Beyond the measures of hardened construction or mobility, some added survivability can be achieved by a dispersed or decentralized design. In some systems, a decentralized design utilizing several data processing centers may be required by economic or other considerations. For example, a single air defense center might be technically feasible for all of NATO, but the communications costs from outlying radar sites would be quite expensive. A completely decentralized design, with a computer center at each radar site, may be equally expensive due to the cost of communications among the centers, but this design is generally more survivable.

When several centers are involved, however, it becomes possible to arrange for a Mode II operation in which one or more centers can "cover" for an adjacent one. Excess data processing capacity and the necessary communication links must be provided.

The last and least attractive possibility in the event of Mode I failure is to revert to a completely manual Mode III operation. Needless to say, the capacity and capability are not attractive, and there is the serious economic problem of maintaining and exercising the added operational personnel.

It is easily seen that Modes I, II, and III cannot be considered independently. The proper selection and design of these modes has a direct bearing on the system configuration and on almost every aspect of the design.

<u>Overhead Facilities</u>

At the early stages of system design and planning, attention should be given to the following non-operational functions:

(a) Training of operational and maintenance personnel.

(b) Initial and on-going program production.

(c) On-going test and evaluation of evolving system changes and additions.

(d) Generation of exercise materials.

(e) Reduction and analysis of data recorded during system operation.

These <u>system</u> support functions -- as opposed to those support functions which must be conducted at each <u>site</u> -- generally require separate "overhead" facilities due to the large computer time requirements and special conditions involved.

The first three functions -- personnel training, program production, and test and evaluation -- require system-like equipments in system-like configurations, although perhaps not with the added equipments (added modules or full duplexing) required for very high reliability and continuity of operation. The last two functions -- generation of exercise materials and data reduction and analysis -- can use system-like equipments, but could also use other computers or commercial computing facilities.

Depending upon the specific nature of the system, one overhead facility might be sufficient to handle the first three functions. In large systems this may not be possible because of the requirements on computer time. In SAGE -- with about twenty-two operational direction centers at peak -- the number of such overhead facilities has varied from three to five, and until this year at least one computer was devoted to each of these functions.

An attractive possibility is to use the overhead machine(s) as a part of the backup or Mode II configuration. In this case and when an operational machine is unavailable, an overhead machine could cease its non-operational functions and join the operational net. This might be considered as a form of duplexing, with the two machines at separate locations.

Overhead facilities, particularly those required for training and program production, assume added importance since they are generally required prior to the operational facilities. An overhead facility for subsequent test and evaluation can be used early in the life of the system for the design verification described in the next section.

DESIGN PROCESS AND TECHNIQUES

Volumes have been written on the subject of the different tools and techniques -- ranging from probability theory to simulation -- which can be employed by the system designer or engineer. How these can best be used and how the design effort should proceed are much more difficult to reduce to writing. Beyond stating that design is both analysis and synthesis, that it must involve feedback, and that it is hardly ever finished, I intend here to mention only two key techniques -- design documentation and design verification -- which have been found to be of practical utility in many systems.

Design Documentation

The need for an early agreement on the requirements and a matching conceptual design has been noted. The design effort starts here and must bring into play full consideration of costs and schedules. Much interplay, much give and take, and much analysis and synthesis may be required. The product should be an operational plan, an employment plan, or some other suitably-named document which will serve as the broad plan or prospectus for the system.

At an early point in the design effort, careful thought should be given to the types and levels of documentation to be used. Documentation is one of the key management tools for the design effort, with regard to both the timeliness and quality of its execution, as well as its conformance to user and other requirements. Documentation, then, is a design technique; it is a key to organizing a design effort and to maintaining design control. The time and effort devoted to generating and maintaining the documentation will be very well spent.

The names of the documents are not important. To quote merely a few: operational specifications, mathematical specifications, system and subsystem performance specifications, functional specifications, system and subsystem design specifications, computer program specifications, and equipment specifications. What is important is that there be recognized levels of documentation and that the responsibilities for each be assigned. Too often, the design is not properly documented and hence not available to those who must be brought to bear on the production and implementation effort when the system has been broken-down into the smaller pieces required for such activities.

As noted, at the highest level there should be an over-all description of what the system will do and how it will be deployed. Next, the over-all performance of the system -- at least in qualitative terms -- should be described, including identification and definition of the principal subsystems. Such system performance specifications might be the vehicle for contracting with a prime contractor. If associate contractors are used, they should be held to meeting subsystem performance specifications. In most cases, the detailed design specification describing what is built will be prepared by the contractors.

In this regard, it should be noted that it is exceedingly important to document a baseline system configuration at the earliest date. This configuration has a first order impact on facilities, construction, civil engineering, and the government-furnished equipments. The documentation of this configuration, ultimately describing all facilities, equipments, computer programs, and technical manuals in the system, must be adequately maintained and all changes controlled.

Design Verification

The second design technique is design verification. By this is meant any reasonable steps which can be taken to validate the adequacy of the design at the earliest possible time. The objective is to avoid costly changes during production or even more costly field retrofits. The latter possibility might involve installation teams waiting unproductively at sites while design problems are diagnosed and solutions generated.

Design verification should start early in the conceptual phase of the design and should be continued, as necessary, until field implementation starts. During this period, the key or critical aspects of the design should be verified, checked, and tested to remove as much uncertainty as possible in the final design. In an air defense system, for example, the tracking logic should be tested under a wide range of realistic conditions; solutions to interception equations should be checked; man-machine relationships and capabilities should be verified; and so on.

Many different methods can be employed for design verification, ranging from complete simulation to live tests with actual equipments. In the former, the computational process, the operators, the environment, and even the system equipments can be simulated; the system computer or some other machine can be used as the vehicle for conducting the simulation; and the entire process can be carried out in a convenient time scale. Live tests, on the other hand, can involve breadboard, prototype, or early production equipments.

Simulation permits an investigation of performance over a wide range of conditions, but generally suffers from the lack of realism. We simulate what we know, not the unexpected, and generally our design has accounted for what we know. Live tests are more difficult and time-consuming, and in some instances are too expensive or even impossible to conduct -- as for example when the test involves many weapons, ECM, etc. In many cases the methods can be mixed or used to supplement one another, as for example the use of selective live tests to calibrate a set of simulations.

The methods employed will vary from system to system. The significant point is that design verification should be considered as an essential part of the design process. It should be incorporated into the planning and funding for the system. It should be applied to the critical areas of the design as early and as realistically as possible. It is one of the few bits of insurance that a system engineer can buy.

HARDWARE

Computers

Significant improvements have been made in the speed, capacity, and reliability of digital computers over the last ten years. Today, many machines of proven performance are available, and they exhibit strong similarities in basic design features: order codes, indexing systems, interrupt systems, and ability to handle auxiliary memory devices (drums, tapes, discs). The chief differences are in word lengths, memory sizes, and operating speeds.

For the most part, computers represent one of the few "off-the-shelf" items that the system designer has at his disposal. There are some related hardware design problems, however, primarily in the area of the special-purpose in-out buffers and devices peculiar to the system. As regards the central computer, it is more a question of proper selection of machine and configuration rather than design.

The critical consideration in the selection of a machine is that of adequate capability: memory capacity and operating speed. It has already been mentioned that care must be taken to allow excess capacity both for contingencies in the original design as well as for unseen requirements. A machine that looks just big enough at the early stages of design is surely not going to be adequate at the end. A safety factor of two would not seem unreasonable, particularly as the cost of more storage and higher speeds seems to be decreasing.

When comparing the required speed of computation against available machines, one should be careful of high computational speeds achieved by special machine features requiring sophisticated software, either in the assembly/compiler programs or in the operational programs. Many machines can only achieve these high speeds when the program structure permits use of the special hardware features. As a more general point, one should not attempt to economize on hardware by assuming efficient, well-written programs -- these are both difficult and costly to achieve. Well-trained, experienced programmers are in very short supply.

The second aspect of machine selection relates to the configuration required to achieve the desired reliability. Until quite recently, if a premium was placed on continuity of operation, a complete duplexing would be required. This was done with considerable success in SAGE. Two machines are available at each direction center, with only one being operational at a time. The machines are connected by a drum through which they can communicate, thereby permitting the second or standby machine to accumulate the dynamic data base required for rapid assumption of responsibility. The high reliability actually achieved from each machine, coupled with a programmed ability to recover from most intermittent errors, has led to very long mean-times between disabling system failures. Several years ago when I last checked, the mean free-time was about 20 days.

Today the computer situation is changing, and the selection of a machine or configuration of machines has taken on several different aspects. This changing situation results from the development of modular machines. In their first appearance, the modularity was restricted to memory capacity and in-out channel capacity. If more of either were needed, added modules could be connected. Today, the modular concept has extended to the central processor itself, and the truly modern machine design includes the capability of employing several processors operating in parallel and sharing the available memory and in-out modules. This permits what has been termed "multi-processing," with several processors operating together on a single job. (It is to be distinguished from "multi-programming," in which one machine works on several different tasks.)

Through multi-processing, modular machines can achieve very high effective operating speeds. However, except for those very few real-time systems which might require operating speeds beyond the capability of a single central processor, the advantages of modularity lie other than in the direction of high speeds.

First, of course, is the capability for growth by addition of modules. Second, is the high reliability which can be achieved at a relatively low price; a full duplex is not needed, rather only one or two spare modules are required for each different element. Further, with proper software design it is possible for individual modules to fail without complete interference with the system operation. This has been termed as a "fail-softly" or "fail-gracefully" characteristic. Third, there is the ability to tailor the equipment -- that is, the number of modules -- to the situation or capacity required at different sites. That is, if a 300-aircraft control system were needed at one site, six memory modules might be required; but at a site requiring only 50 tracks, only two modules would be needed.

Parallel operation of computers is employed in the NTDS or Navy Tactical Data System in which a three-processor configuration is required to do the full fleet air defense task on a major ship. One machine does tracking, another does the intercept calculations, and a third processes display information and performs miscellaneous tasks. Under less than full-load conditions, two machines can be used with different programs to perform the same job at lower capacity. The third machine is then available for maintenance or other support tasks. It is also possible to operate at a one-machine level, and this is done on smaller ships not requiring intercept control capabilities.

The NTDS design, however, does not utilize a common memory; rather the individual computers exchange information via in-out channels. The

French STRIDA II air defense system -- for which only limited design information is available -- employs multiple computers, each permanently assigned to specific tasks but sharing some common memory.

The full, multi-processing potential of the recent modular machines has not yet been realized in actual systems. The FAA design for an improved air traffic control system -- the National Airspace System -- will ultimately incorporate a full modular multi-processing capability, but initial system implementation will be along the more conventional lines.

The software design problems associated with exploiting a full multi-processing capability are quite significant. They include the problem of breaking down the program into small, relatively independent parts and the design of adequate executive routines to handle the traffic, to sense modular malfunctions, and to manage the assignments and switching of modules. There are related hardware problems. It may be some years yet before it is desirable (or necessary) to spend the money and effort to solve these problems as long as the more conventional high-speed sequential machines in a simple duplex configuration are adequate to the tasks.

Display Consoles

The situation with display consoles is quite the opposite from that of computers. There are relatively few "off-the-shelf" equipments, there have been only limited advances in performance or cost over the past ten years, and there are few systems in which the user is satisfied with his display consoles.

The situation is aggravated by the fact that it is extremely difficult to reach an agreement on requirements. The display console is one part of the system in which the operator normally takes a strong interest, and he naturally desires the utmost in performance. Unfortunately, however, he is an easy prey of the "brochuremanship" technique, since he does not always appreciate the difference between a paper design and a working model, or between a laboratory model and a field-maintained production unit. The net result, then, is that his requirements are very high: much information, rapidly changing, alpha-numeric characters, flicker-free presentation, a bright display under high illumination levels, and possibly even color.

The designer finds it difficult to challenge the need, and the problems of complexity, cost, and maintainability are not received sympathetically by the operator. It is unfortunate that it is not generally possible to quickly put together and demonstrate various display capabilities so that the advantage or need of various features could be objectively determined before proceeding with production equipments.

Display consoles, then, represent a most difficult problem for the designer. While it may be impossible to completely satisfy the operator, he must be given a useable and reliable display. Economic factors cannot be ignored, particularly at the present production costs of $30,000 to over $100,000 per console. Added points of caution or consideration include:

(a) Whenever possible, a standard console design should be adopted, with very minor modifications for specific operating positions.

(b) Character or symbol sizes should be kept as small as possible within the limits of legibility. Special provisions (perhaps in the software) may be required to prevent overlap of symbology, as might result from adjacent aircraft tracks.

(c) The ambient lighting environment should be carefully understood or designed, particularly as this may cause reflections on the scope face.

(d) The general requirement for large viewing surfaces should be balanced against smaller viewing areas coupled with a capability for off-centering and expansion.

(e) The merits of alpha-numeric characters as opposed to a limited symbolic capability should be matched against the differences in equipment complexity and cost.

(f) In addition to the console-operator interface, attention should be given to the computer-console interface. The display console is usually one of the earliest identified subsystems, but its design must carefully consider the interface with the computer, and particularly with the computer program. The tradeoff between the amount of computer programming involved in display formatting and the complexity of the console itself is not an easy one to make.

(g) Requirements for background or geography displays must be considered.

Large Screen Displays

The situation with large screen displays is not very different from that of consoles in that requirement are difficult to resolve, the user generally adds a multiple-color requirement, and the available systems are limited. A large number of techniques ranging from dry processes to wet processes and from zerographic techniques to theater TV techniques have been proposed, but only a few have yet reached the stage of working systems.

At the present time, silver halide systems involving projection of images photographed from the face of a CRT are available with processing times of about ten seconds. The maintenance problems of such systems seem under control. Four color systems using separate projectors or filters are used in a number of systems; color mixing has not generally been reduced to field practice.

Beyond reliability, a major problem is that of brightness, since the command posts in which these projected displays are used are normally kept at high illumination levels.

Input Devices

Special-purpose action switches and general-purpose keyboards are the principal devices by which operators insert data into the system or influence the processing. Switches are quick and easy to use, but pose a problem when many different possible actions are required. In order to retain the simplicity of switch inputs while keeping the number of switches within reasonable bounds, some recent designs have utilized switches capable of performing several different functions by either manual or automatic label changes.

For inputs requiring more flexibility, particularly those involving variable-length items or alpha-numeric messages, a keyboard is required. The problems of format and content errors have resulted in sophisticated equipments, generally termed "message composers," to assist the operator. Other designs have relied upon the computer to help the operator in the composition-formatting-error detection-error correction process by a feedback of computer confirmation or error information on printers. Standard teletype machines can then be used as the input device.

For operators working on consoles with plan position displays, an ability to designate or point out selected positions can be achieved by a photoelectric cell "light gun" or "light pencil" or by a movable display circle or "hook" controlled by a "tracking ball" (or "joy-stick"). Both of these types of devices can also be used as a more general input technique whereby the operator may select and designate quickly with his "pointer" among a number of alternatives presented in an alpha-numeric message form on the display console. This technique, an extension of some earlier work on "electronic type-writers," appears to have considerable promise in command systems requiring rapid, lengthy, and flexible man-machine exchanges.

SOFTWARE

The computer programs, or software, are of central importance since they direct the data processing equipment, and hence the operation of the system. They usually represent a sizeable fraction of the total system design and development effort, and their cost in most systems is comparable to that of the computers themselves.

Nevertheless, the results achieved in this area leave much to be desired. The universal experience is that the magnitude of the computer programming activity, both in time and effort, is grossly underestimated. Further, the inherent potential of these programs for ease of modification has not been realized; in practice, and for a variety of reasons, the operational programs have not been flexible, and to change them has been costly and time-consuming.

A first design problem, then, is to recognize the total size and scope of the programming task. In addition to the operational program itself -- which, as noted earlier, should contain performance monitoring, data recording, checkout features, etc. -- it is necessary to plan for the other programs required in the software production, checkout, test, and installation. These additional programs can be categorized as follows:

(1) Utility programs necessary for fabricating the operational program. These cover such areas as

assembly, diagnostics, tape handling and processing, analysis, documentation, and control.

(2) Support programs used to expedite the testing of the operational program during fabrication. These would provide the parameter and other data inputs required for such program testing phases as parameter testing of subprograms, assembly testing of groups of sub-programs, program shakedown, and system shakedown.

(3) The test data reduction programs required to evaluate the performance of the operational program during test phases.

(4) The operational data analysis programs required to support evaluation of system operation on a longer-term basis.

The number and size of the programs in these categories vary with each different system. In SAGE, which pioneered much of the work on the types of utility, support, and data reduction programs needed for real-time systems, a very large effort was expended on the non-operational programs. Further, the rates at which programs in the different categories can be designed and produced vary significantly.

Production rates on the operational program, in particular, may be an order of magnitude less than on conventional scientific and engineering programs. Because of its size, the operational program must usually be broken down into smaller pieces for individuals to work on. This complicates both the design problem and the subsequent assembly and checkout of the various subprograms. Because the subprograms may refer to or change common items of stored information and because it is not always easy or desirable to freeze the design of the data tables at an early stage, special techniques -- notable the use of common symbolic tags and compools -- have been established. This, then, requires that sophisticated special-purpose compilers must be used. Further, since real-time data processing generally requires exploiting the speed of the machine, a good deal of machine-language programming may be required.

The production rates of operational programs are further lowered by the extensive checkout required. Each subprogram must be tested under a variety of input conditions -- this is often called parameter testing -- and then the programs must be assembled together and checks made on the continuity of operation. Finally, the entire program should be tested under a wide variety of operating conditions.

To illustrate these points, Table I shows some estimates on program sizes and efforts for a proposed real-time data processing system for which considerable related experience is available. All programs were assumed to be produced with a modified assembly program, with the exception of the bulk of the data reduction and analysis programs which would be written for a separate commercial machine using FORTRAN. The "production rate" includes all activities beginning with the program design activities (given program performance specifications) and terminating with the handover of a tested program, including card decks, listings, design and coding specifications, and manuals.

In this table, it was assumed that the computer came with the normal repetoire of assembly, loader, trap, and trace programs, and that the special-purpose equipment test programs required to check out and test various equipment subsystems -- display consoles, input/ output equipments, data links, etc. -- would be provided by equipment suppliers.

Table I demonstrates two points of common experience:

(1) The size of the supporting programs is generally greater than the operational program by a factor of 2 to 5.

TABLE I: PROGRAM PRODUCTION ESTIMATES

CATEGORY	SIZE (Single Address Instructions)	PRODUCTION RATE (Instructions Per Man-Month)	PRODUCTION EFFORT (Man-Months)
Operational	50,000	70	700
Utility	40,000	250	160
Support	40,000	250	160
Test Data Reduction	120,000 20,000	1200 250	180
Operational Data Analysis	30,000	250	120
TOTALS	300,000		1320

(2) The effort in producing the supporting programs generally equals and may exceed the effort on the operational program.

As an added word of caution, it should be noted that estimators of program sizes are traditionally optimistic (they base their estimates on what they think they themselves could do), yet the program is inevitably written largely by relatively new and unproductive programmers (experienced programmers generally graduate to writing sophisticated compilers or they become managers).

The key lessons in this size and effort area are:

(1) Identify and plan for all necessary computer programs at the earliest date.

(2) Do not underestimate the checkout and documentation activities.

(3) Do not assume program production rates normally achieved in scientific or business data processing programs.

(4) Expect reduced production rates as the magnitude of the operational program and number of subprograms grow.

A second major software problem is the organization of the operational program into relatively independent modules or subprograms. For example, should display generation functions be distributed among the various subprograms which may affect displays -- tracking, identification, weapons assignment, etc. -- or should they be grouped into one display generation subprogram? Economy of storage and operating time points to consolidation; flexibility for change points to distribution.

Once the functions of each subprogram module are determined, it becomes necessary to define the precise inputs and outputs -- that is, the transfer function of the module.

The fixed and dynamic data storage tables offer many design choices. Here, too, efficiency of storage utilization usually runs counter to the design which offers the greatest flexibility.

A master or executive program is normally required to direct the sequential execution of subprograms, to handle transfers of tables and other data from the high speed memory to and from other storage media, and to handle the in/out and interrupt processing. The executive program must be capable of handling subprograms operating at different rates; some are required periodically, some only on demand. The executive program must also possess high flexibility for addition of new subprograms and for modifications of program sequence or periodicity. In summary, it merits the most careful design, including consideration of the instrumentation and testing requirements imposed during program assembly and checkout.

Throughout the software design, attention must be given to the need for and techniques of adapting a master operational program for use at different sites with individual characteristics or parameters.

During the design process, attention should also be given to the matter of response time, and, in particular, to the elapsed time from an operator input or request to the related computer output or response. Three times are involved: recognition of the input and routing to the proper subprogram, subprogram processing, and the final output processing. Proper design of the terminal equipment can minimize the initial and final steps. Some difficulty has been experienced in several systems with the subprogram processing. Several subprograms may be involved, and if searching of lengthy files is required, surprisingly long response time -- on the order of minutes -- can result. Priority processing of inputs and careful, and possibly redundant, table design may be required.

As noted, ease and rapidity of modification represent outstanding software design problems. The current lead-times of months, and possibly up to a year, for modest field changes are clearly undesirable and tend to belie the name of software. With the availability of larger and faster machines, it is becoming possible to design the program into relatively independent, although less efficient, modular packages. If this is done, new program modules can be added and old program modules changed without requiring a major modification of the entire program.

A very interesting possibility, adapted in part from business data processing, is the development of general-purpose programs or software. This concept would exploit the similarity of many functional processes in the operational programs, particularly in command systems involving data manipulation. For example, the functions of file updating, file retrieval, message processing, display make-up, or report generation could be programmed in a very general form, and then adapted to specific applications by supplying the detailed descriptors for the files and the data. An extension of this technique would then permit <u>on-line</u> compilation of programs by operational personnel to achieve specific needs. Initial research in these areas appears to offer promising results, although not without large requirements on computer storage and operating time.

Finally, the initial program production effort must be carried out with due regard to the subsequent production effort. The general practice in the United States is for assumption of these on-going efforts by military personnel. This places stringent demands on the documentation of the initial program, may influence the choice between problem-oriented vs machine-oriented compilers, and requires integration of key military personnel in the initial design and production activity.

TESTWARE AND TESTING

Throughout the system design and engineering effort, adequate attention must be given to the testware--the plans, equipments, computer programs, and procedures and techniques associated with the system test activities.

Early definition of the phases of the system test activity is essential. These phases might include design verification, the so-called Category I and II tests relating to equipment acceptance and test at the manufacturer's plant and an initial field site, implementation tests at successive field sites, a full-scale operational evaluation, and follow-on experimental tests leading to improvements.

Again the names are not significant, and the nature and type of testing will vary from system to system. The significant point is that a decision as to the nature and types of test activity be made at a sufficiently early date to permit a determination of the requirements for special equipments and facilities, special computer programs, test teams, operational personnel, special flight tests, computer time, etc. This should then be followed up with the necessary plans, schedules, contracts, and other arrangements.

The manpower requirements for planning, designing, documenting, conducting, and analyzing a test program in a large system can be sizeable and suitable provisions should be made. The use of a separate contractor not associated with the design or production of hardware or software has considerable merit from the point of view of objectivity. It will also go a long way toward assuring that sufficient attention is given to the entire testware problem. Such a separate and independent test contractor was used in SAGE with considerable success.

Of particular importance is an early recognition that the test plans and schedules must take into account certain facts of system life. Not all tests will be conducted as originally planned; tests will be delayed and disrupted for a variety of reasons; and tests will uncover errors and deficiencies that will require substantial amounts of time for redesign and correction. Optimistic test schedules are not realistic.

In a multi-site system or one involving many remote input-output locations or connections, the sequential phasing of the test activity requires special attention. In all systems, a carefully-generated, methodical plan for the availability and test of the subsystems and various sites is critical. To the greatest degree possible, there should be a capability to test and verify subsystem performance independently of the rest of the system. For this purpose, instrumentation--in the form of equipments and programs--must be provided.

The test activity is only meaningful if there are criteria against which measured performance can be checked and if the system inputs and environment can be controlled when the performance is measured.

It is necessary to determine what is to be measured, under what conditions this is to be done, and how much data to be collected (the size of the sample). None of this is easy to do. Finally, the level of acceptable performance must be decided. This is a particularly difficult but necessary task. It generally requires much compromise and agreement among the designer, tester, and ultimate user.

Verification of test procedures and determination of the test measures should be conducted as early as possible, and can be one of the objectives of a design verification activity. In a multi-site system, Category II tests can provide the performance measures against which successive sites will be checked.

In designing the test activity, consideration must be given to the availability of trained operational personnel. Without adequately-trained operators, it may be impossible to conduct useful tests. A corollary problem, of course, is the early availability of trained maintenance personnel.

Finally, conducting tests at a site while maintaining manual operations may pose severe problems. In an air defense system, for example, it is extremely difficult to share the use of search, beacon, and height-finding radars. These potential problems--and suitable operational procedures or equipment modifications--also require early consideration.

All in all, the experience to date has been that adequate provisions for the testware are not usually made, resulting in grossly-extended and costly test periods. The technical problems of defining adequate test criteria and obtaining system inputs that are sufficiently controlled (or at least known) to allow meaningful measurements are not yet solved. We still do not expend sufficient time and effort in planning and preparing for the test activity well before it starts.

CONCLUSION

It is perhaps too early for an objective assessment of the development of the first generation of military real-time data processing systems. From the limited perspective of participating engineers and designers, this paper has attempted to summarize the key system engineering lessons of this experience.

It must be recognized that these lessons are not necessarily unique to military real-time data processing systems, but may generally be true of all large system endeavors. This realization, in fact, may be the most significant conclusion we could reach.

Beyond that point, however, I should like to add further emphasis to several items:

(1) It is important to develop a proper appreciation of the full magnitude and scope of the effort at the start. The system engineer must take a very broad point of view, far beyond the confines of operational equipments and programs. In particular, the design must make provisions for a wide range of essential support functions and activities at the site and system level.

(2) The management structure and assignment of responsibilities for system acquisition can have a profound effect on the final product. Operational representation and inputs are essential;

strong central design control is required regardless of contractual mechanisms and responsibilities.

(3) Outstanding system design problems are the proper matching of man/machine capabilities and the provision of adequate capacity and flexibility for change and growth.

(4) Documentation and design verification are vital elements of the design process.

(5) Software problems are invariably underestimated, and much remains to be done to realize the inherent potential of these programs for ease and rapidity of modification.

(6) Testware design merits comparable attention with hardware and software.

Finally, and perhaps needless to say, system engineering is a necessary function in the system development and acquisition process. It does not "just happen"; it must be carefully planned and deliberately applied.

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) The MITRE Corporation Bedford, Massachusetts	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

SYSTEM DESIGN AND ENGINEERING FOR REAL-TIME MILITARY DATA PROCESSING SYSTEMS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

N/A

5. AUTHOR(S) (Last name, first name, initial)

Israel, David R.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
January 1965	68	0

8a. CONTRACT OR GRANT NO. AF 19(628)-2390 b. PROJECT NO. 416L c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TDR-64-168 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) SR-124

10. AVAILABILITY/LIMITATION NOTICES

Qualified requestors may obtain from DDC.
DDC release to OTS authorized.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	416 L/M/N System Program Office, L. G. Hanscom Field, Bedford, Massachusetts

13. ABSTRACT

This report treats the key problems and considerations arising in the design, engineering, and implementation of military systems in which real-time data processing plays a central role. The principal distinguishing characteristics of these command and control systems are summarized. Organizational matters relating to responsibilities, operational inputs, and procurement aspects are described in the context of the over-all system acquisition process. Initial considerations which should guide the over-all design are discussed, including such outstanding design problems as the proper matching of man/machine capabilities and the provision of adequate capacity and flexibility for change and growth. Important aspects of hardware, software, and testware design are also detailed.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Data Processing Systems (Military)						
Real-Time						
Design and Engineering						
Command and Control Systems						
Real-Time						
Data Processing, Design for Military Applications						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

________________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

________________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

________________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional